Extending partial combinatory algebras.

Bethke, I.; Klop, J.W.; de Vrijer, R.C.

published in
Mathematical Structures in Computer Science (MSCS)
1999

DOI (link to publisher)
10.1017/s0960129599002832

document version
Publisher's PDF, also known as Version of record

Math. Struct. in Comp. Science (1999), *vol.* 9, *pp.* 483–505. Printed in the United Kingdom

Extending partial combinatory algebras

INGE BETHKE[†], JAN WILLEM KLOP[‡] and ROEL de VRIJER[§]

[†]*University of Nijmegen, Department of Computer Science*
Email: inge@cs.vu.nl

[‡]*University of Nijmegen, Department of Computer Science*
and
CWI, Amsterdam
and
Vrije Universiteit Amsterdam, Department of Computer Science
Email: jwk@cwi.nl

[§]*Vrije Universiteit Amsterdam, Department of Computer Science*
Email: rdv@cs.vu.nl

Received 15 September 1998; revised 16 April 1999

We give a negative answer to the question of whether every partial combinatory algebra can be completed. The explicit counterexample will be an intricately constructed term model. The construction and the proof that it works depend heavily on syntactic techniques. In particular, it provides a nice example of reasoning with elementary diagrams and descendants. We also include a domain-theoretic proof of the existence of an incompletable partial combinatory algebra.

1. Introduction

Consider a structure $\mathfrak{A} = \langle A, s, k, \cdot \rangle$, where A is some set containing the distinguished elements s, k, equipped with a binary operation $\cdot$ on A, called application, which may be partial.

Notation 1.1.

1 Instead of $a \cdot b$ we write ab; and in writing applicative expressions, the usual convention of association to the left is employed. So for elements $a, b, c \in A$, the expression $aba(ac)$ is short for $((a \cdot b) \cdot a) \cdot (a \cdot c)$.
2 $ab \downarrow$ will mean that ab is defined; $ab \uparrow$ means that ab is not defined. Obviously, an applicative expression can only be defined if all its subexpressions are.
3 If t_1, t_2 are applicative expressions, $t_1 \cong t_2$ means that either both $t_1 \uparrow$ and $t_2 \uparrow$, or $t_1 \downarrow$ and $t_2 \downarrow$ and $t_1 = t_2$.

Definition 1.2. A structure $\mathfrak{A}$ as indicated above is a *partial combinatory algebra* (pca) if for all $a, b, c \in A$:

1 $ka \downarrow$, $sa \downarrow$, $sab \downarrow$,
2 $kab \cong a$ and $sabc \cong ac(bc)$.

If, moreover, the application operator is total, that is, if $ab \downarrow$ for all $a, b \in A$, then $\mathfrak{A}$ is called *total*, or just a *combinatory algebra* (ca).

Note that instead of $kab \cong a$, we can equivalently write $kab = a$. In general, we have $t_1 = t_2$ whenever $t_1 \cong t_2$ and $t_2 \downarrow$.

Examples 1.3.

1 A well-known example of a pca is Kleene's $\mathfrak{K} = \langle \mathbb{N}, s, k, app \rangle$, where app is defined as the Kleene-bracket application from recursion theory: $app(m, n) \cong \{m\}(n)$, and s and k are appropriately chosen to satisfy the characterizing axioms. (See Example 5.6.5 in Mitchell (1996).) As a matter of fact, there are (infinitely) many possible choices for s, k, with each choice yielding an alternative variant of $\mathfrak{K}$.

2 Another example is given by the Uniformly Reflexive Structures of Strong (1968) and Wagner (1969).

3 The 'initial' pca is obtained within Combinatory Logic (CL). Here one takes all closed, strongly normalizable CL-terms modulo convertibility by means of the S- and K-axioms. Application is defined iff the result is again strongly normalizing. In every pca all applicative expressions corresponding to strongly normalizing CL-terms are defined. In particular, all normal forms are defined.

4 One might also think that the weakly normalizing CL-terms modulo CL-convertibility constitute a pca, with application defined iff the result is again weakly normalizing, by analogy with the pca of SN-terms in (3). However, this is not the case. For, consider $\omega = SII$ with $I = SKK$. Then $S \cdot K \cdot \omega \cdot \omega$ is defined, but $K \cdot \omega \cdot (\omega \cdot \omega)$ is undefined.

5 On the other hand; there is an interesting class of CL-terms that we will call PN, *persistently normalizable* CL-terms, for which the construction in (4) does yield a pca. Define $M \in PN$ iff every subterm of a reduct of M has a normal form. If SN is the set of strongly normalizable CL-terms and WN the set of weakly normalizable CL-terms, we have SN $\subseteq$ PN $\subseteq$ WN. An example of a term in PN - SN is

$$[\mathrm{S(K(KI))(SII)}][\mathrm{S(K(KI))(SII)}].$$

6 Asperti and Ciabatoni (1995; 1996) have introduced 'effective applicative structures' (eas); an eas is equivalent to a pca. See Remark 8.1 for a description of the notion of an eas.

Remark 1.4.

1 In proper pca's (*i.e.,* nontotal pca's) we have $sk \neq ki$ with $(i = skk)$. For, if $sk = ki$, it would follow from $skab \cong kb(ab)$ and $kiab \cong ib \cong b$ that every ab is defined.

2 Likewise, in every proper pca we have for all $a \in A$ that $k(ka) \neq s(k(ka))$. Otherwise it would follow from

$$k(ka)bc \cong kab \cong a$$

and

$$k(ka)bc = s(k(ka))bc \cong k(ka)b(bc) \cong ka(bc)$$

that $bc \downarrow$, for all $b, c \in A$.

The question we address in this paper is whether it will always be possible to extend a partial combinatory algebra to a total combinatory algebra by, if needed, supplementing the domain with new elements, and completing the application operation on the extended domain. Formally, the notion of extension is given by the following definition.

Definition 1.5. An *extension* of a partial combinatory algebra $\mathfrak{A} = \langle A, s, k, \cdot_A \rangle$ is a partial combinatory algebra $\mathfrak{B} = \langle B, s, k, \cdot_B \rangle$ (same s and k), such that $A \subseteq B$ and $\cdot_A = \cdot_B | \mathrm{dom}(\cdot_A)$, that is, $\cdot_A$ coincides with the restriction of $\cdot_B$ to $\mathrm{dom}(\cdot_A)$.

So the question is whether every pca has a total extension. It was raised by H.P. Barendregt, G. Mitschke and D. Scott and included in the list of open problems at the Swansea lambda calculus meeting of September 1979, which was organized by Roger Hindley. We quote from Hindley (1980):

> This question was originally asked about the time of the Swansea 1974 meeting, and it seems both important and difficult. Whatever the answer, some interesting mathematics will probably result.

The negative answer was announced in Klop (1982), in a short note. The present paper elaborates that announcement, along the lines of the sketch given there.

2. Heuristics

It was observed as early as Hindley (1980) that the following straightforward attempt to complete a pca $\mathfrak{A} = \langle A, s, k, \cdot \rangle$ fails: add a new element $*$ to A, and extend $\cdot$ to $A \cup \{*\}$ by defining $ab = *$, if $ab \uparrow$ in $\mathfrak{A}$, and $*a = a* = ** = *$.

The reason this does not work is that for all $a \in A$, we then have $a = ka* = *$.

Another reason why this fails is that in the pca of Example 1.3(3) (of the strongly normalizable CL-terms modulo convertibility) it is inconsistent to equate all not strongly normalizable terms. See, for example, Barendregt (1984).

Next, one might try to proceed by adding 'formal elements' $\underline{ab}$ whenever $ab \uparrow$, extending the application to such a and b by stipulating $ab = \underline{ab}$, and then dividing out the 'appropriate' equivalence relation. However, as our proof of the existence of incompletable pca's will show, no such procedure can be uniformly successful.

It is well known that one can formulate a 'non-erasing' version CL_I of CL, using instead of S and K the basic combinators I, J, satisfying $Ia = a$, $Jabcd = ab(adc)$. Also, the well-known combinators B, C, I, W can be used for a non-erasing version of CL. It is obvious how to formulate the corresponding notion of 'non-erasing' pca, with distinguished elements i, j (or b, c, i, w). Now it is not hard to show that for such non-erasing pca's based on $\{i, j\}$ or $\{b, c, i, w\}$ with corresponding rules there is no problem in extending to a total ca along the lines mentioned above.

We will now describe the intuition behind our syntactic construction of a pca $\mathfrak{A}$ that cannot be completed. In completing a pca, a previously undefined expression kt_1t_2 may become equal to a previously defined expression t_1 by virtue of the k-equation

$$kt_1t_2 = t_1 \ .$$

Now suppose that a pca could be devised in such a way, that after any would-be

completion we would be forced to have $kt_1t_2 = ks_1s_2$, where s_1 is again a previously defined expression, but such that in the original pca $t_1 \neq s_1$. Then the assumption that a completion exists would necessarily yield an inconsistency:

$$t_1 = kt_1t_2 = ks_1s_2 = s_1 \ .$$

A counterexample pca where this indeed happens can be realized as follows. Consider a pca $\mathfrak{A}$, containing distinct elements a, b, c, and such that

$$s(sk)a = s(sk)b \ .$$

Then we have as follows a conversion between ac and bc:

$$\begin{aligned} s(sk)ac &= s(sk)bc \\ skc(ac) &= skc(bc) \\ k(ac)(c(ac)) &= k(bc)(c(bc)) \\ ac &= bc \ . \end{aligned}$$

Now suppose we can arrange that in $\mathfrak{A}$ we have $ac\downarrow$, $bc\downarrow$, $ac \neq bc$, but $c(ac)\uparrow$, $c(bc)\uparrow$. The above conversion between ac and bc will then be ruled out since it involves the undefined expressions $c(ac)$ and $c(bc)$. We will also arrange that other conversions between ac and bc will be ruled out; they 'essentially' amount to the one above and will contain subterms 'essentially' the same as the forbidden $c(ac)$ and $c(bc)$. Such a pca $\mathfrak{A}$ will be incompletable. For in any completion $\mathfrak{A}^*$ of $\mathfrak{A}$ the conversion between the distinct elements ac and bc would go through.

As a guideline for the construction of a pca $\mathfrak{A}$ as sketched above, one can think of a, b, c as 'bearers of undefinedness'. In small doses, isolated or in an application such as ac, bc they are harmless, but the presence in an expression of 'large' clusters of them, like $c(ac)$, where large means 'length $\geqslant 3$', make the expression undefined. Also, expressions that reduce, in the usual Combinatory Logic sense, to undefined expressions, will be undefined. (So, for example, $\omega(\omega a)$, where $\omega = sii$, is an undefined expression, since it reduces to the large cluster $aa(aa)$.)

The construction just sketched will be performed within Combinatory Logic. In order to construct elements a, b, c with the required properties, some new constants A, B and C will be added to the combinators S and K. Note, also in view of the completability of non-erasing pca's (see above), that the 'culprit' is the K-combinator, or, in the pca, the axiom for k, with its erasing effect.

In Figure 1 we have summarized the situation. The terms AC, BC are convertible using the axioms for S, K (the downward arrows) and the axiom $S(SK)AC = S(SK)BC$ (the horizontal transition). However this conversion is not valid as it leads through 'forbidden territory', namely through the area $\mathscr{U}$ of undefined expressions ($\mathscr{D}$ is the area of defined expressions). This area $\mathscr{U}$, the shaded, cone-like part of the figure, is an absolute barrier; every attempted conversion between the two terms AC, BC must pass the forbidden area. The forbidden area contains all expressions that reduce to the utmost forbidden terms on the bottom of the cone, namely those terms containing a 'large cluster', among them $C(AC)$ and $C(BC)$.

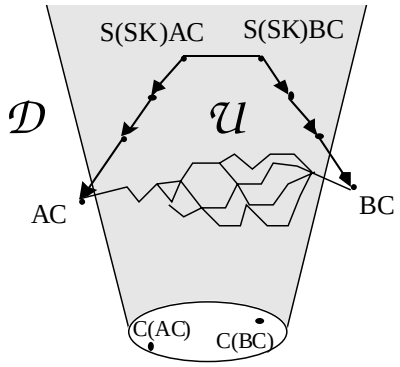

Fig. 1. Incompletable pca

3. Combinatory Logic and traces in CL conversions

In this section we collect the necessary basic properties of Combinatory Logic reduction. We emphasize the use of elementary diagrams in constructing Church–Rosser diagrams, and introduce the notion of 'trace'.

3.1. *Descendants*

We start with the classical notion of descendants and ancestors in CL. Definitions of this notion can be found in Curry and Feys (1958) and Barendregt (1984). Here we will give an 'algebraic' definition, using a labelled version of CL. In fact, our definition works in general for orthogonal first-order TRSs, of which CL is an example. Our definition is taken from Bethke *et al.* (1997a) and Bethke *et al.* (1997b).

Let $L = \{\alpha, \beta, \gamma, \ldots\}$ be a set of *labels*. They can be thought of as colours that will be used to keep track of symbols. A term is *labelled* by equipping some of its symbol occurrences with a label as superscript. A labelling of a term t can be seen as a partial function from the set of symbol occurrences in t to L. If this function is I, we also denote the labelled term by t^I. A labelling I is *initial* if it is total and injective. This means that in t^I all symbol occurrences have different labels.

Let R be an orthogonal first-order TRS (see, for example, Baader and Nipkow (1998) and Klop (1992)). Let R have a rewrite rule, for example,

$$\rho : F(G(x, y), H) \to S(T(y), y) \ .$$

Then a *labelled version* of ρ has the form

$$F^{\alpha}(G^{\beta}(x, y), H^{\gamma}) \to S(T(y), y)$$

for some $\alpha, \beta, \gamma \in L$. That is, every occurrence of a function symbol in the *redex pattern* $F(G(\ ,\ ), H)$ is equipped with a label as superscript; in the *contractum pattern* $S(T(\ ),\ )$ no symbol is labelled.

Now the *simply*† *labelled version* R_L of R is defined by taking as terms the terms of R where each symbol may have a label, and as rewrite rules all labelled versions of the rules of R.

Example 3.1.1. Specializing this definition to CL, we have the following rules for the labelled system CL_L (for all $\alpha, \beta, \gamma, \delta \in L$):

$$\begin{aligned} @^{\alpha}(@^{\beta}(@^{\gamma}(S^{\delta}, x), y), z) &\to @(@(x, z), @(y, z)) \\ @^{\alpha}(@^{\beta}(K^{\gamma}, x), y) &\to x\ . \end{aligned}$$

Here the prefix symbol @ is used for application.

Proposition 3.1.2.

1 If R is an orthogonal TRS, then its simply labelled version R_L is also orthogonal.
2 If t^I is a labelling of t, then each reduction step $t \to s$ in R can be *lifted*, in a unique way, to a reduction step $t^I \to s^J$ in R_L.

We can now define descendants in reductions in R.

Definition 3.1.3. Let $t \to s$ be a reduction step in R. Give t an initial labelling t^I and assume that lifting the step $t \to s$ results in the labelled step $t^I \to s^J$. Now let p^{α} be a labelled symbol occurrence in t^I and q^{α} in s^J, for some label $\alpha \in L$. Then we say that $p \in t$ is the *ancestor* of $q \in s$, or that q is the *descendant* of p, or that p, q are in the *ancestor–descendant* relation.

In Klop (1980) the notation $p \dashrightarrow q$ is used to denote the fact that q descends from p. We will also say that p *traces* to q, or that q can be *traced back* to p.

Note that in a step $t \to s$, every symbol q in s has a unique ancestor symbol in t. Instead of defining the descendant relation between symbol occurrences, we could equivalently have defined the descendant relation between subterm occurrences. The equivalence is seen by noting that symbol occurrences are in 1–1 correspondence with subterm occurrences, namely by taking the top (the root) of the subterm. Traditionally, descendants of redex occurrences are also called *residuals*. Note that the redex contracted in $t \to s$ has no descendants.

Now that we have defined the descendant relation for single steps $s \to t$, it is obvious how to define the descendant relation for many step reductions $s \twoheadrightarrow t$ by transitivity. The transitive descendant relation is denoted by $\dashrightarrow\!\!\!\!\!\!\rightarrow$. So, for example, in the two-step reduction $s \to r \to t$, we have for $p \in s$ and $q \in t$ that $p \dashrightarrow\!\!\!\!\!\!\rightarrow q$ if there is an intermediate symbol v in r such that $p \dashrightarrow v \dashrightarrow q$.

Remark 3.1.4. There is a small subtlety here. We could have defined $\dashrightarrow\!\!\!\!\!\!\rightarrow$ for many-step reductions as follows. Give t an initial labelling t^I and lift the reduction $t \twoheadrightarrow s$ to the labelled reduction $t^I \twoheadrightarrow s^J$. Now define, as before, that a symbol p in t traces to q in s iff its label (in t^I) is the same as the label of q (in s^J). So the difference is that in the former

† We use the term *simply* labelled in order to distinguish this notion of reduction from more sophisticated variants such as those introduced by Levy (1975) and others.

definition, tracing is defined by repeated initialization of the labels: in each step the labels are 'refreshed' to an initial labelling. Fortunately, we can without much effort prove that both ways yield the same result. In other words, repeated initialization is superfluous. (A closer look shows that the reason is that labelled reduction stays preserved under not necessarily injective renaming of labels.)

We will now prove that the descendant relation satisfies the property of 'right cancellation'.

Proposition 3.1.5. Consider a reduction $s \twoheadrightarrow r \twoheadrightarrow t$, with symbol occurrences $p \in s$, $v \in r$, $q \in t$ such that $p \dashrightarrow q$ and $v \dashrightarrow q$. Then $p \dashrightarrow v$.

Proof. The proof follows immediately from transitivity of the trace relation and unicity of the ancestor: suppose $v \in r$ traces back to $p' \in s$. Then, by transitivity of 'trace', $q \in t$ also traces back to $p' \in s$. Since q also traces back to $p \in s$, and as ancestors are unique, we have $p = p'$. Hence $p \dashrightarrow q$. □

Remark 3.1.6. *labelling (colouring) preserves the descendant relation.* By this we mean the following. Consider a reduction $s \twoheadrightarrow t$, with symbol occurrences $p \in s$, $q \in t$, such that $p \dashrightarrow q$. Lift this reduction to the labelled reduction $t^I \twoheadrightarrow s^J$ with I initial. Then the symbol occurrences p, q will be in the labelled version p^α and q^α. Now these coloured symbols p^α, q^α are also in the descendant relation in the *labelled* TRS. This is not as trivial as it may look at first sight, but it is not hard to convince oneself of this fact.

In the construction of an incompletable pca in this paper we want to be able to trace symbols not only along a reduction, but through an arbitrary conversion. The ensuing notion of *connectedness* extends the ancestor–descendant relation.

Definition 3.1.7.

1 The relation is the symmetric closure of $\dashrightarrow$.
2 Consider a conversion $\Gamma : s_0 - s_1 - \cdots - s_n$. Here each occurrence of '–' stands for either $\leftarrow$ or $\rightarrow$. A *trace* through Γ is a sequence $p_0 \dashrightarrow p_1 \dashrightarrow p_2 \dashrightarrow \cdots \dashrightarrow p_n$, with $p_i \in s_i$. The symbol occurrences p_0 and p_n are said to be *connected* by this trace.

Example 3.1.8. In the conversion $xz(yz) \leftarrow Sxyz \rightarrow xz(yz)$ we have that all the occurrences of z in the two end terms $xz(yz)$ are connected to each other, via the z in $Sxyz$.

3.2. *Elementary diagrams*

The construction of elementary diagrams emerges from a standard proof of the weak Church–Rosser property (WCR) for CL or any other orthogonal term rewriting system. Two diverging one-step reductions are made to converge by contracting the residuals of the originally contracted redexes.

In the construction of reduction diagrams below, for which the elementary diagrams are the building stones, it is essential that we also take into account any so-called 'empty steps'.

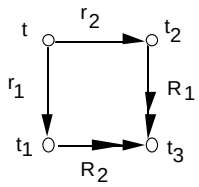

Fig. 2. Elementary diagram

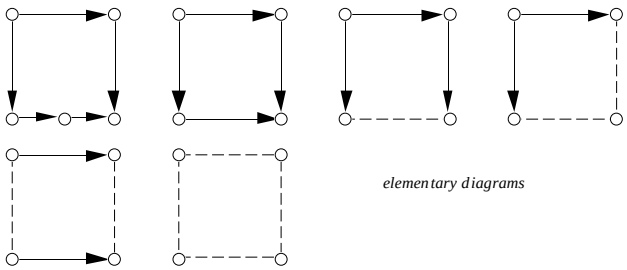

Fig. 3. Improper elementary diagrams

Definition 3.2.1.

1 A *proper elementary reduction diagram* (e.d.) with initial point t is determined as follows. Let r_1, r_2 be redexes in t, such that contraction of r_i results in the reduction step $t \to t_i$ $(i = 1, 2)$. Then the corresponding e.d. has these reduction steps as left-hand side and upper side, respectively. The right-hand side of the e.d. consists of contracting the set R_1 of the residuals of r_1 after the r_2-step (performed, say, in left to right order). The lower side is defined dually. (See Figure 1)

2 If R_1 (or R_2) is empty, as may happen, for example, when the contraction of one of the redexes r_1, r_2 erases the other, then the corresponding side is a so-called *empty step*. We then have $t_3 \equiv t_2$ (or $t_3 \equiv t_1$).

3 A special case of (2) is when r_1 and r_2 coincide. Then, by the definition of residuals, both R_1 and R_2 are empty, and the e.d. is completed with two empty steps.

4 Improper e.d.'s arise by the interaction of empty steps and proper (or again empty) steps in the obvious way. See Figure 4.

An example of an e.d. is given in Figure 3.2. It belongs to CL with the extra constant I and the corresponding rule $Ix \to x$. We call the term in the left-hand upper corner the 'initial' point of an e.d.; the 'final' point is the term in the right-hand lower corner.

For CL we have e.d.'s of the types shown in Figure 4. We indicate the sides that consist of an empty step by dashed lines, which can be seen as reductions of length zero. So, if s - - - - t is an empty step, we have $s \equiv t$.

We extend the definition of the descendant–ancestor relation to the case of empty steps s - - - - t: the relation is simply the identity relation. In the following we will allow

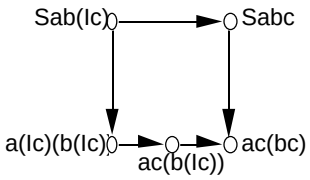

Fig. 4. CL-example of an elementary diagram

conversions to contain empty steps too. It is obvious how to extend the notion of trace, Definition 3.1.7, to conversions with empty steps.

Remark 3.2.2. *Colourability of elementary diagrams.* Consider an e.d. in an orthogonal TRS. Label the initial point with an initial labelling. We can now extend this initial colouring through the whole e.d. by lifting the reductions that constitute the e.d. Then there is the following question to consider. The e.d. consists of a lower reduction and an upper reduction from initial to final point. In lifting these reductions, why is the final term uniquely coloured? It is conceivable that lifting via the upper reduction and lifting via the lower reduction results in differently coloured final terms.

To see that this discrepancy cannot occur, we invoke Remark 3.1.6. That observation gives us that the redexes contracted in the right-hand side of the coloured e.d. are actually descendants (in the coloured TRS!) of the redex contracted in the left-hand side, and likewise dually. This is because they were descendants in the original uncoloured TRS, and colouring was seen to preserve the descendant relation.

But then these coloured lifted upper and lower reductions constitute, by definition, an e.d. in the coloured TRS. Of course any e.d. in an orthogonal TRS, coloured or not, has a unique final point, and hence so also does this coloured e.d. That means that the original e.d. was 'colourable', without the *a priori* possible discrepancy.

The next proposition states a non-trivial property concerning the descendant relation of elementary diagrams in orthogonal TRSs.

Proposition 3.2.3. Let q be a symbol occurrence in the final point of an e.d. Then the ancestor of q in the initial point of the e.d. traced back via the lower side of the e.d. coincides with the ancestor via the upper side.

Proof. The proof is immediate from Remark 3.2.2, stating the colourability of e.d.'s, and the definition of the descendant relation using colours, and using Remark 3.1.4. □

3.3. *Reduction diagrams*

We now turn to building 'reduction diagrams' composed of the e.d.'s of the previous section in a 'paving' process, with the goal of finding a common reduct for terms obtained by different diverging reductions. An example is given in Figure 5. From proofs of the

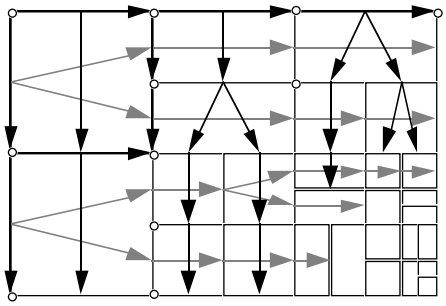

Fig. 5. Paving process

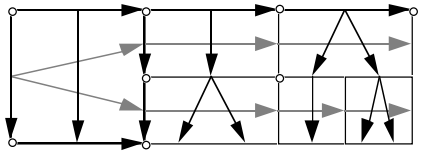

Fig. 6. Parallel moves

Church–Rosser theorem for CL, or, more generally, for orthogonal TRSs, we know that such constructions of 'Church–Rosser diagrams' will indeed terminate successfully. (See, for example, Klop (1980).) In the figure, the e.d.'s contain some transverse arrows that are intended to indicate the 'propagation' of residuals of the various contracted redexes through the diagram. If we set out one reduction step against a reduction of arbitrary length, the situation that arises by completing the diagram is known as the 'Parallel Moves Lemma' (PML). Figure 6 gives an example. The rightmost side of the diagram consists of a reduction of all descendant redexes of the original redex contracted in the downward step. These descendants are disjoint, hence the name 'parallel moves'. We will use PML below, in the proofs of Propositions 5.1 and 5.2. Next, consider a diagram construction that has not yet been finished but is in an intermediate stage (Figure 7). We will be especially interested in the border of such an unfinished diagram, the staircase-like part. It is a conversion in CL, possibly interlaced with empty steps.

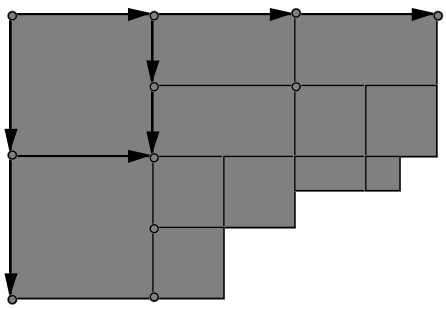

Fig. 7. Unfinished diagram construction

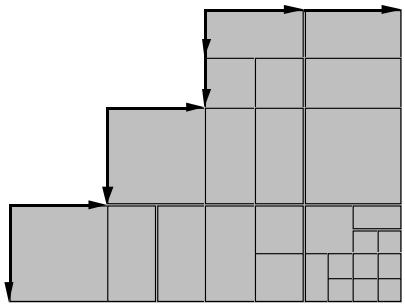

Fig. 8. Completion of an arbitrary conversion

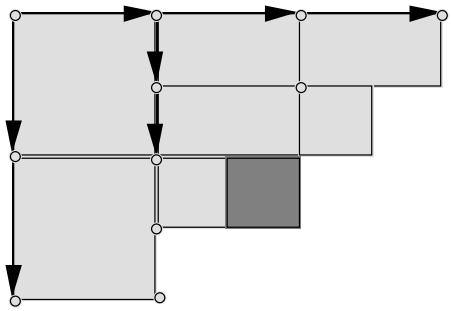

Fig. 9.

Remark 3.3.1. We have described above how a completed reduction diagram (a 'Church–Rosser diagram') arises by tiling with e.d.'s, starting from an initial configuration formed by two diverging reductions. We should point out that we can start with an arbitrary conversion (a 'staircase') as initial configuration, and similarly be assured of successful completion. Figure 8 illustrates this situation.

3.4. *Trace lifting*

Definition 3.4.1. Consider (in Figure 9) an unfinished diagram, the lightly shaded one, with a border Γ. Let the border after adjoining the darker shaded e.d. be Γ'. Then we say that the conversion Γ is *higher* than Γ, notation $\Gamma \Rightarrow \Gamma'$. We will denote the transitive-reflexive closure of this relation with the same notation. Note, in particular, that in a completed reduction diagram (as in Figure 8), the initial conversion (the staircase) is higher than the final conversion of the converging reductions (lower and right-hand side).

Proposition 3.4.2. Consider an e.d. as in Figure 10. Let symbols $q \in t$, $q' \in t'$ be connected via the conversion $t \twoheadrightarrow r \leftarrow t'$ (the lower side of the diagram). Then they are also connected via $t \leftarrow s \rightarrow t'$ (the upper side of the diagram).

Proof. Consider the trace connecting q and q' through the lower side. Say it intersects r in v. Now let $p \in s$ be the unique ancestor of $v \in r$; by Proposition 3.2.3 it is independent

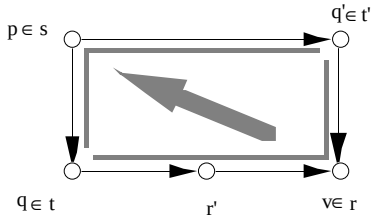

Fig. 10.

of whether it is found via the upper or the lower reduction of the e.d. Now, applying the right cancellation property (Proposition 3.1.5), we find that p is connected to t', and likewise to t. Hence q and q' are connected via the upper half of the e.d., as we sought to prove. □

Proposition 3.4.3. (Trace lifting). Consider two conversions Γ and Γ' of intermediate construction stages in a completed reduction diagram such that $\Gamma \Rightarrow \Gamma'$. Let s, t be the common endpoints of both conversions. Moreover, let $p \in s$, $q \in t$ be symbol occurrences connected via a trace through Γ'. Then p and q are also connected via the higher conversion Γ.

Proof. The proof is immediate from the corresponding fact for e.d.'s (Proposition 3.4.2), the construction of diagrams, and the definition of 'higher'. □

Remark 3.1. The converse of trace lifting does not hold. A trace through a conversion cannot always be pushed down to a trace through a lower conversion. A counterexample has already been shown in Example 3.1.8. Consider the e.d. with two identical diverging steps $Sxyz \to xz(yz)$ and hence two empty converging steps. Along the upper conversion, consisting of the diverging steps, all z's are connected. But this is not the case along the lower conversion, consisting of the empty steps.

4. Definition of the counterexample

Definition 4.1.

1 Let $X = \{A, B, C\}$ be a set of three new constants and let CL_X be Combinatory Logic extended with this set: that is, the equational system having

$$\mathrm{Ter}(\mathrm{CL}_X) = \{M \mid M \text{ is built by application from } S, K, A, B, C\}$$

as set of terms, and equipped with the usual axioms for S and K.

2 $\mathrm{CL}_X \vdash M = N$ means that the terms M, N are convertible by means of the S- and K-axiom.

3 We will also consider S- and K-reduction and write $\mathrm{CL}_X \vdash M \twoheadrightarrow N$ when M can be reduced to N using the reduction rules

$$\begin{aligned} Sxyz &\to xz(yz) \\ Kxy &\to x. \end{aligned}$$

Definition 4.2. To the equational system CL_X we add the axiom

$$S(SK)A = S(SK)B \ .$$

The result will be denoted as CL_X^*, and $\mathrm{CL}_X^* \vdash M = N$ denotes convertibility (or derivability) in the extended system.

Next we define the crucial notion of *relativized convertibility*.

Definition 4.3. If Σ is an equational calculus and $\mathscr{D} \subseteq \mathrm{Ter}(\Sigma)$, then

$$\Sigma \vdash_{\mathscr{D}} M = N$$

iff there is a conversion $M = M' = M'' = \cdots = N$ between M and N in Σ, such that $\{M, M', M'', \ldots, N\} \subseteq \mathscr{D}$.

Definition 4.4.

1 If $M \in \mathrm{Ter}(\mathrm{CL}_X)$ contains only A, B, C, we will call M a *cluster*. A cluster is *large* if its length is $\geqslant 3$, where the length of a cluster is defined inductively by $length(M) = 1$ if $M \in \{A, B, C\}$, and $length(M_1 M_2) = length(M_1) + length(M_2)$ otherwise.
2 The set $\mathscr{D} \subseteq \mathrm{Ter}(\mathrm{CL}_X)$ is defined as follows:

$$M \in \mathscr{D} \Leftrightarrow \text{no } N \text{ such that } \mathrm{CL}_X \vdash M \twoheadrightarrow N \text{ contains a large cluster} \ .$$

The set $\mathscr{D}$ is to be thought of as the set of defined expressions.

Remark 4.5.

1 $\mathscr{D}$ is closed under reduction.
2 The complement of $\mathscr{D}$, the set $\mathscr{U}$ of undefined terms, is closed under expansion.
3 If $M \in \mathscr{D}$ and M' results from M by replacing an occurrence of a symbol from X by another, say A by B, then $M' \in \mathscr{D}$ also.

Observations (1) and (2) are immediately clear.

As for (3), if $M' \notin \mathscr{D}$, then there is a large cluster L' such that $M' \twoheadrightarrow C[L']$. Reversing the replacement and carrying this through for the complete reduction would then result in a reduction $M \twoheadrightarrow C[L]$, with L a large cluster, contradicting the assumption that $M \in \mathscr{D}$.

We will now define the pca $\mathfrak{A}$ that cannot be completed.

Definition 4.6.

1 Define $\mathfrak{A}$ as the pca that has as universe the set A of $\sim$-equivalence classes $[M]$ of terms, where $\sim$ is defined by

$$M \sim N \Leftrightarrow \mathrm{CL}_X^* \vdash_{\mathscr{D}} M = N \ .$$

(Note that $M \sim N \Rightarrow M, N \in \mathscr{D}$.)
2 We put $s = [S]$, $k = [K]$, $a = [A]$, $b = [B]$, $c = [C]$.
3 Application in $\mathfrak{A}$ is defined as follows. If $x = [X]$, $y = [Y]$, then

$$x \cdot y = \begin{cases} [XY] & \text{if } XY \in \mathscr{D}, \\ \uparrow & \text{otherwise} \ . \end{cases}$$

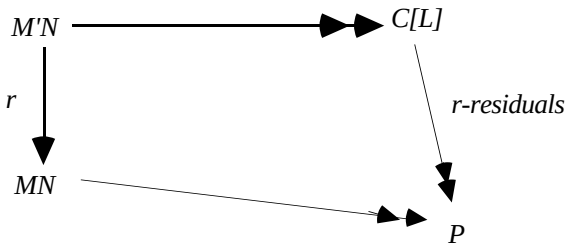

Fig. 11.

5. Proof that $\mathfrak{A}$ is a pca

We have first to make sure that application is well defined, *i.e.*, independent of the choice of the representing terms M, N. This is implied by the following proposition.

Proposition 5.1. Let $M \sim M'$ and $N \sim N'$. Then $MN \in \mathscr{D} \Rightarrow M'N' \in \mathscr{D}$.

Proof. We will only consider a one-step conversion in one of the terms M, N. Then the proposition follows by induction on the total number of steps in the two conversions given by $M \sim M'$ and $N \sim N'$. So assume $MN \in \mathscr{D}$. We now consider only the case that the one-step conversion takes place in M; the reasoning for N will be the same. So we have the assumption that $M \sim M'$ by a one-step conversion and must verify that $M'N \in \mathscr{D}$. There are three cases to consider.

1 $M \to M'$: Then $MN \to M'N$, and $M'N \in \mathscr{D}$ follows since $\mathscr{D}$ is closed under reduction, by Remark 4.5(1).
2 $M \sim M'$ by an application of $S(SK)A = S(SK)B$: Then $M'N \in \mathscr{D}$ follows by Remark 4.5(3).
3 $M' \to M$: Assume for a proof by contradiction, that $M'N \notin \mathscr{D}$. Then $M'N \twoheadrightarrow C[L]$ for some context $C[\ ]$ containing a large cluster L. The Parallel Moves Lemma now yields the diagram in Figure 11, with $P \in \mathscr{D}$ because it is a reduct of MN. The reduction from $C[L]$ to P consists of contractions of residuals of the redex r, and, since $P \in \mathscr{D}$, this reduction erases the large cluster L. Such erasure is only possible by a K-step, say by contraction of the redex $r' = KQ(\cdots L \cdots)$. Since r' is a residual of r, we must have $r = KQ'L'$, where $L' \twoheadrightarrow (\cdots L \cdots)$. But then $M' \notin \mathscr{D}$, contradicting the assumption that $M \sim M'$. □

In order to see that with $\mathfrak{A}$ we have indeed defined a pca, Clauses 1 and 2 of Definition 1.2 must be verified.

Proposition 5.2.

1 For all $x, y \in A$ we have $kx\downarrow$, $sy\downarrow$, $sxy\downarrow$.
2 In $\mathfrak{A}$ the equations $kxy \cong x$ and $sxyz \cong xz(yz)$ hold for all $x, y, z \in A$.

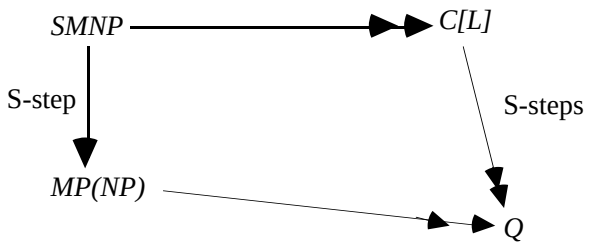

Fig. 12.

Proof.

1 We consider $sxy\downarrow$, the other two cases being even more obvious. It amounts to showing: $M \in \mathscr{D}, N \in \mathscr{D} \Rightarrow SMN \in \mathscr{D}$. Note that all reducts of SMN are of the form $SM'N'$, with $M \twoheadrightarrow M'$ and $N \twoheadrightarrow N'$. A large cluster in $SM'N'$ should lie within either M' or N', contradicting the assumption that $M, N \in \mathscr{D}$.

2 Obviously, $KMN \in \mathscr{D} \Leftrightarrow M, N \in \mathscr{D}$, and hence for $M, N \in \mathscr{D}$ we have $KMN \sim M$ by an application of the K-rule; so $kxy = x$.
Likewise, verifying the s-axiom boils down to proving that $SMNP \in \mathscr{D} \Leftrightarrow MP(NP) \in \mathscr{D}$. One direction, ($\Rightarrow$), is obvious, since $\mathscr{D}$ is closed under reduction. For ($\Leftarrow$), suppose $SMNP \notin \mathscr{D}$. So there is a context $C[\,]$ and a large cluster L, such that $SMNP \twoheadrightarrow C[L]$. Now the Parallel Moves Lemma can be used again as in Figure 12. It follows, since S-steps are non-erasing, that Q must contain at least one occurrence of L. Hence $MP(NP) \notin \mathscr{D}$. □

6. Proof that $\mathfrak{A}$ is incompletable

We need only to execute the plan that was outlined in Section 2. That is, the result that $\mathfrak{A}$ is not completable follows from the following proposition.

Proposition 6.1.

1 In $\mathfrak{A}$ we have $ac\downarrow$, $bc\downarrow$, $c(ac)\uparrow$, $c(bc)\uparrow$ and $ac \neq bc$.
2 In every supposed completion $\mathfrak{A}^*$ of $\mathfrak{A}$, however, $ac = bc$.

Proof. Part (2) is already covered by the conversion given in Section 2 (there for a, b, c):

$$\begin{aligned} S(SK)AC &= S(SK)BC \\ SKC(AC) &= SKC(BC) \\ K(AC)(C(AC)) &= K(BC)(C(BC)) \\ AC &= BC\ . \end{aligned}$$

Of part (1), the first four assertions follow at once from the definitions. For example, $c(ac)\uparrow$ because $C(AC)$ is a large cluster.

In order to prove $ac \neq bc$, we must show that $\mathrm{CL}^*_X \not\vdash_{\mathscr{D}} AC = BC$. To do this, we will show that every conversion between AC, BC in CL^*_X will necessarily contain undefined terms, *i.e.,* terms that reduce to a term containing a large cluster. (For the shortest

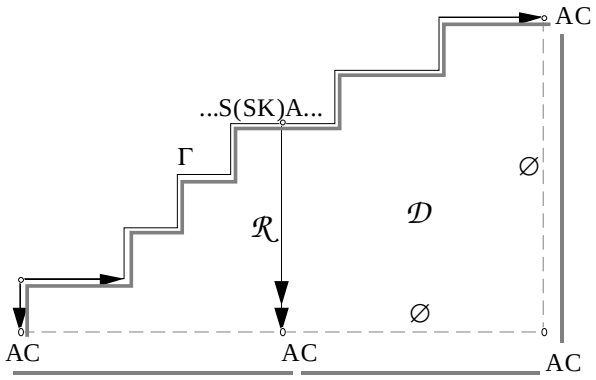

Fig. 13.

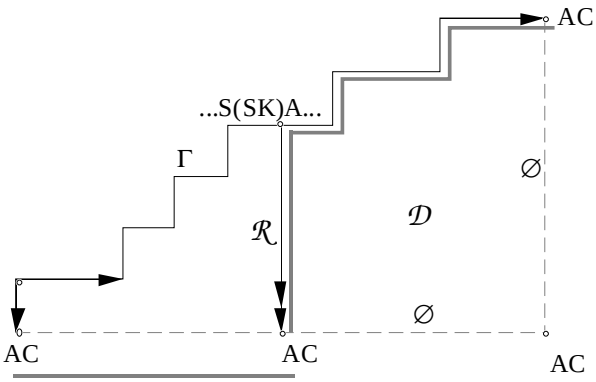

Fig. 14.

conversion, as given above, this is seen to hold at once: except for AC, BC every term in it is not an element of $\mathscr{D}$.) So let Γ^* be an arbitrary CL^*_X-conversion $AC = \cdots = BC$ for a proof by contradiction. Replace every B in Γ^* by A. Since a step $S(SK)A = S(SK)B$ is transformed into an empty step, the result is a CL_X-conversion $\Gamma : AC = \cdots = AC$. By Church–Rosser and the fact that AC is a normal form, we have the diagram in Figure 13. Obviously, the A's in AC and AC are connected via the empty reductions ($\varnothing$). Since $\Gamma \Rightarrow \cdots \Rightarrow \varnothing$, by the trace lifting Proposition 3.4.3, we have that the A in the first term of Γ can be traced to the A in the last term. Now consider the trace $A \dashrightarrow A \dashrightarrow A \dashrightarrow \cdots \dashrightarrow A$ through Γ. Restoring the original B's, we have a 'pseudo trace' $A \dashrightarrow \cdots \dashrightarrow B$; hence a step $A \dashrightarrow B$ where A changes to B. This can only happen in a step in Γ^* of the form $C[S(SK)A] = C[S(SK)B]$, for some context $C[\ ]$. (In the Figures 13, 14 the term $C[S(SK)A]$ is rendered as $...S(SK)A...$.)

So in Γ one of the terms is of the form $C[S(SK)A]$. In the diagram of Figure 13 we obviously have (by the procedure of constructing reduction diagrams) a reduction $\mathscr{R} : C[S(SK)A] \twoheadrightarrow AC$, as shown in Figure 14. Now in pushing up the lower trace (through the empty reductions), it is clearly possible to do this in such a way that one of

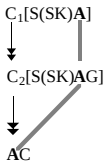

Fig. 15.

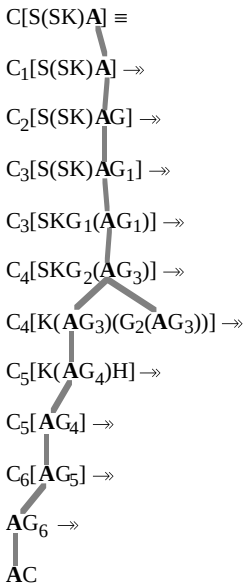

Fig. 16.

the intermediate stages gives a trace through $\mathscr{R}$ as in Figure 14. Since $\mathscr{R}$ is a reduction, this means that the A in $C[S(SK)A]$ is in fact the ancestor of the A in AC. Now we note that A is *passive* in $S(SK)A$ and *active* in AC[†]. By some simple arguments it is easily shown that such a change in status entails that $\mathscr{R}$ must have the form as in Figure 15. And some further consideration shows easily that $\mathscr{R}$ has the more detailed form of Figure 16, where there is a trace through the occurrences of A displayed as $\boldsymbol{A}$. Here $G \twoheadrightarrow G_1 \twoheadrightarrow G_2$, $G_1 \twoheadrightarrow G_3 \twoheadrightarrow G_4 \twoheadrightarrow G_5 \twoheadrightarrow G_6 \twoheadrightarrow C$, $G_2(AG_3) \twoheadrightarrow H$, and $C_i[\,] \twoheadrightarrow C_{i+1}[\,]$. Hence, by Church–Rosser, we also have $G_2 \twoheadrightarrow C$, and thus $G_2(AG_3) \twoheadrightarrow C(AC)$, which is a large cluster. Hence several of the terms in $\mathscr{R}$ are not in $\mathscr{D}$: in particular, the first one, $C_1[S(SK)A]$, which is a term in Γ isn't. But then Γ^* also contains a term not in $\mathscr{D}$ (by

[†] In an application MN the term M is in active position, N in passive position.

$D \triangleleft [D \to D]$	Plotkin (1985)
$D \triangleleft [D_\perp \to_\perp D_\perp]$	Bethke (1987)

Table 1. *Domain equations*

Remark 4.5(3)). However, Γ^* was a CL^*_X-conversion, which gives a contradiction. Hence $ac \neq bc$. □

7. Domain-theoretic considerations

In this section we will briefly indicate how an incompletable pca can be constructed by domain-theoretic considerations. It boils down to the incompletability of all so-called extensional pca's that are nontotal.

It is well known that domains D satisfying one of two conditions in Table 1 lend themselves to the construction of pca's. Here

1 $A \triangleleft B$ denotes that A is a retract of B;
2 $[D \to D]$ is the set of partial continuous functions from D to D, that is, functions whose domain of definition is a Scott-open subset of D, and whose restriction to this subset is a (total) continuous function;
3 $[D_\perp \to_\perp D_\perp]$ is the set of continuous functions from $D_\perp$ (that is, D equipped with a bottom element) to $D_\perp$ that preserve the bottom element.

In particular, domains D satisfying either $D \cong [D \to D]$ or $D \cong [D_\perp \to_\perp D_\perp]$ give rise to extensional pcas. As stipulated in the following definition, in an extensional pca elements with the same functional behaviour are equal.

Definition 7.1. A pca $\mathfrak{A}$ is *extensional* if for all $a, a' \in A$ we have

$$(\forall a'' \in A\ a \cdot a'' \cong a' \cdot a'') \Rightarrow a = a' .$$

These extensional pcas, if they are nontotal, can be easily shown to be incompletable (Bethke 1987; Bethke and Klop 1996).

Theorem 7.2. Let $\mathfrak{A}$ be an extensional, nontotal pca. Then $\mathfrak{A}$ is incompletable.

Proof. Suppose $\mathfrak{A} = \langle A, s, k, \cdot \rangle$ is such a pca and assume $\mathfrak{A}' = \langle A', s, k, \cdot' \rangle$ is a completion of $\mathfrak{A}$. Let $\perp \in A$ be some totally undefined element – for instance, one constructed as follows: take $a, a' \in A$ such that $aa' \uparrow$ and put $\perp = s(ka)(ka')$. Then, indeed, $\perp a'' \uparrow$ for every $a'' \in A$. So we have $s(k(kk))\perp a'' \uparrow$ and $s(k(ks))\perp a'' \uparrow$ for every $a'' \in A$. By extensionality, we therefore have $s(k(kk))\perp = s(k(ks))\perp$, and hence $s(k(kk))\perp k = s(k(ks))\perp k$. But then in $\mathfrak{A}'$ we have

$$\begin{aligned} s = ks(\perp k) = k(ks)k(\perp k) &= s(k(ks))\perp k \\ & s(k(kk))\perp k = k(kk)k(\perp k) = kk(\perp k) = k , \end{aligned}$$

which gives a contradiction. □

Remark 7.3.

1 Note that in fact '$\perp$-extensionality' (*i.e.,* equality of all nowhere defined elements) is already a sufficient condition for incompletability.
2 Whereas the proof of the incompletability of extensional nontotal pca's consists of just a few lines, showing the existence of such structures is a laborious matter that requires no less effort than the syntactic construction and proofs above (see, for example, Bethke (1987)).

8. Concluding remarks and questions

Remark 8.1. *A weaker notion of completability.* In Asperti and Ciabatoni (1997) the authors show that satisfaction of Barendregt's axiom[†] guarantees a weaker notion of completability, namely the one where application can be extended to a total operation with the constants s and k possibly chosen anew. This already follows from Asperti and Ciabatoni (1996) where it was proved that the *unique head-normal forms condition*, which is sufficient for completability (Bethke *et al.* 1996), can be equivalently expressed by the effective and injective realization of the recursion-theoretic s-m-n theorem.

The precise formalization of this property relies on an alternative characterization of pca's as suitable collections of partial functions introduced in Asperti and Ciabatoni (1995) under the name 'Effective Applicative Structures' (eas). An eas is a collection of indexed partial functions that is closed under composition, contains all projections and an interpreter, and satisfies the s-m-n theorem. In Asperti and Ciabatoni (1996) it was, moreover, proved that the injectivity of the realization is equivalent to the so-called 'Padding Lemma'. Since Barendregt's axiom implies the lemma (see Barendregt (1975)), one gets as a corollary that this axiom is enough to ensure weak completability.

Remark 8.2. It is also possible to avoid the use of *extraneous constants* A, B, C as above and to give an incompletable pca whose whole universe is generated by $\{S, K\}$ alone. This can be done by defining

$$\begin{array}{lcll} A^* & = & SII(SII) & \text{with} \quad I = SKK \\ B^* & = & SI'I'(SI'I') & \text{with} \quad I' = SKS \\ C^* & = & SI''I''(SI''I'') & \text{with} \quad I'' = SK(SK)\,. \end{array}$$

In this way, A^*, B^*, C^* behave sufficiently like the inert constants A, B, C (they are terms of order zero, and are not able to interact with the context); the arguments involving tracing become more complicated though, because A^*, B^*, C^* may still exhibit some internal activity.

Remark 8.3. *Curry's equations.* As noted in Remark 1.4, a pca satisfying $sk = ki$ must be complete; likewise when the pca satisfies $k(ka) = s(k(ka))$ for some a. In fact, the equation $sk = ki$ is derivable from Curry's equations A_β (Barendregt 1984). For, $(SK)_\lambda =_\beta (KI)_\lambda$ as a simple verification shows. Theorem 7.3.10 in Barendregt (1984) now states that $\mathrm{CL} + \mathrm{A}_\beta \vdash SK = KI$. This does not yet imply that a pca satisfying A_β is already complete, since the conversion between SK and KI in $\mathrm{CL} + \mathrm{A}_\beta$ could employ undefined subterms. However, one can exhibit an actual derivation $\mathrm{CL} + \mathrm{A}_\beta \vdash SK = KI$ that only employs defined subterms. Hence we arrive at the following fact.

[†] Barendregt's axiom (Barendregt 1975, Definition 1.1.7) is: $sab = sa'b' \Rightarrow a = a' \;\&\; b = b'$.

Theorem 8.4. $\mathfrak{A} \models \mathrm{A}_\beta \Rightarrow \mathfrak{A}$ is total.

Proof. As explained above, we have $\mathrm{CL} + \mathrm{A}_\beta \vdash SK = KI$. We need, moreover, an *actual* derivation in which only defined subterms occur. Now both sides of all axioms in A_β are normal forms, and hence are always defined. The use of the S-axiom is always 'safe' too. So, we need an actual derivation in which the applications of the K-axiom do not erase or introduce undefined terms. Such a derivation was given by Wojtek Swiatek, and is included in the Appendix. It consists of 44 steps; and it is easily checked that the only terms erased or introduced by the K-axiom are normal forms, and hence are defined. In fact these terms are K, I, KI and $S(KS)(S(KK)K)$. □

Appendix

The material in this appendix was kindly made available to us by Wojtek Swiatek.

Let $\boldsymbol{CL}$ be the following set of axioms:

$\boldsymbol{K}PQ = P$
$\boldsymbol{S}PQR = PR(QR)$
$P = P$
$P = Q \Rightarrow Q = P$
$P = Q, Q = R \Rightarrow P = R$
$P = Q \Rightarrow PR = QR$
$P = Q \Rightarrow RP = RQ$

Let $\boldsymbol{A}_\beta$ be the following set of axioms:

(A.1) $\boldsymbol{K} = \boldsymbol{S}(\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{KK})\boldsymbol{K}))(\boldsymbol{K}(\boldsymbol{SKK}))$
(A.2) $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{K}(\boldsymbol{S}(\boldsymbol{KS})))(\boldsymbol{S}(\boldsymbol{K}(\boldsymbol{S}(\boldsymbol{KK})))\boldsymbol{S})))(\boldsymbol{K}(\boldsymbol{K}(\boldsymbol{SKK})))$
(A.3) $\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{KK})(\boldsymbol{S}(\boldsymbol{KS})\boldsymbol{K})))(\boldsymbol{KK}) = \boldsymbol{S}(\boldsymbol{KK})$
(A.4) $\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{KK})) = \boldsymbol{S}(\boldsymbol{KK})(\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{KK})(\boldsymbol{SKK}))))(\boldsymbol{K}(\boldsymbol{SKK}))$
(A.5) $\boldsymbol{S}(\boldsymbol{K}(\boldsymbol{S}(\boldsymbol{KS})))(\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{KS}))) = \boldsymbol{S}(\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{KK})(\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{K}(\boldsymbol{S}(\boldsymbol{KS})))\boldsymbol{S}))))(\boldsymbol{KS})$

We will show that in $\boldsymbol{CL}$ with $\boldsymbol{A}_\beta$ it is true that $\boldsymbol{SK} = \boldsymbol{KI}$, by a conversion that only erases normal forms. Let

$\boldsymbol{I} \equiv \boldsymbol{SKK}$,
$F \equiv \boldsymbol{S}(\boldsymbol{KS})$,
$G \equiv \boldsymbol{S}(\boldsymbol{KK})$,
$D \equiv \boldsymbol{S}(F(G\boldsymbol{I}))(\boldsymbol{KI})$.

Hence the axioms $(A.1), \ldots, (A.5)$ can be written as:

(A.1) $\boldsymbol{K} = \boldsymbol{S}(F(G\boldsymbol{K}))(\boldsymbol{KI})$
(A.2) $\boldsymbol{S} = \boldsymbol{S}(F(\boldsymbol{S}(\boldsymbol{KF})(\boldsymbol{S}(\boldsymbol{KG})\boldsymbol{S})))(\boldsymbol{K}(\boldsymbol{KI}))$
(A.3) $\boldsymbol{S}(F(G(F\boldsymbol{K})))(\boldsymbol{KK}) = G$
(A.4) $FG = GD$
(A.5) $\boldsymbol{S}(\boldsymbol{KF})(FF) = \boldsymbol{S}(F(G(F(\boldsymbol{S}(\boldsymbol{KF})\boldsymbol{S}))))(\boldsymbol{KS})$

Note that for all terms P and Q we have:

$FPQ \equiv \boldsymbol{S}(\boldsymbol{KS})PQ = \boldsymbol{KS}Q(PQ) = \boldsymbol{S}(PQ)$,
$GPQ \equiv \boldsymbol{S}(\boldsymbol{KK})PQ = \boldsymbol{KK}Q(PQ) = \boldsymbol{K}(PQ)$.

We now have

$$
\begin{aligned}
\boldsymbol{SK} &= \boldsymbol{S}(\,\boldsymbol{S}(F(G\boldsymbol{K}))(\boldsymbol{KI})\,) && (A.1)\\
&= \boldsymbol{S}(\,\boldsymbol{S}\,(F(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S})))\,(\boldsymbol{K}(\boldsymbol{KI}))\,(F(G\boldsymbol{K}))\,(\boldsymbol{KI})) && (A.2)\\
&= \boldsymbol{S}(F(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S}))(F(G\boldsymbol{K}))\,(\boldsymbol{K}(\boldsymbol{KI})(F(G\boldsymbol{K})))\,(\boldsymbol{KI})) && (CL)\\
&= \boldsymbol{S}(\,F(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S}))(F(G\boldsymbol{K}))\,(\boldsymbol{KI})(\boldsymbol{KI})) && (CL)\\
&= \boldsymbol{S}(\,\boldsymbol{KS}(F(G\boldsymbol{K}))\,(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S})(F(G\boldsymbol{K})))(\boldsymbol{KI})(\boldsymbol{KI})) && (CL)\\
&= \boldsymbol{S}(\boldsymbol{S}\,(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S})(F(G\boldsymbol{K})))\,(\boldsymbol{KI})(\boldsymbol{KI})) && (CL)\\
&= \boldsymbol{S}(\boldsymbol{S}(\,\boldsymbol{K}F(F(G\boldsymbol{K}))\,(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S}(F(G\boldsymbol{K}))))(\boldsymbol{KI})(\boldsymbol{KI})) && (CL)\\
&= \boldsymbol{S}(\boldsymbol{S}(F\,(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S}(F(G\boldsymbol{K}))))\,(\boldsymbol{KI})(\boldsymbol{KI})) && (CL)\\
&= \boldsymbol{S}(\boldsymbol{S}(F(\,\boldsymbol{K}G(F(G\boldsymbol{K}))\,(\boldsymbol{S}(F(G\boldsymbol{K})))))(\boldsymbol{KI})(\boldsymbol{KI})) && (CL)\\
&= \boldsymbol{S}(\,\boldsymbol{S}(F(G(\boldsymbol{S}(F(G\boldsymbol{K}))))(\boldsymbol{KI})(\boldsymbol{KI})\,) && (CL)\\
&= \boldsymbol{S}(F(G(\boldsymbol{S}(F(G\boldsymbol{K}))))(\boldsymbol{KI})\,(\boldsymbol{KI}(\boldsymbol{KI}))\,) && (CL)\\
&= \boldsymbol{S}(\,F(G(\boldsymbol{S}(F(G\boldsymbol{K}))))(\boldsymbol{KI})\,\boldsymbol{I}) && (CL)\\
&= \boldsymbol{S}(\,\boldsymbol{KS}(\boldsymbol{KI})\,(G(\boldsymbol{S}(F(G\boldsymbol{K})))(\boldsymbol{KI}))\boldsymbol{I}) && (CL)\\
&= \boldsymbol{S}(\boldsymbol{S}\,(G(\boldsymbol{S}(F(G\boldsymbol{K})))(\boldsymbol{KI}))\,\boldsymbol{I}) && (CL)\\
&= \boldsymbol{S}(\boldsymbol{S}(\,\boldsymbol{KK}(\boldsymbol{KI})\,(\boldsymbol{S}(F(G\boldsymbol{K}))(\boldsymbol{KI})))\boldsymbol{I}) && (CL)\\
&= \boldsymbol{S}(\boldsymbol{S}(\boldsymbol{KK})\boldsymbol{I}) && (A.1)\\
&= \boldsymbol{KSI}\,(\boldsymbol{S}(\boldsymbol{KK})\boldsymbol{I}) && (CL)\\
&= \boldsymbol{S}\,(\boldsymbol{KS})\,(\boldsymbol{S}(\boldsymbol{KK}))\,\boldsymbol{I} \equiv FG\boldsymbol{I} && (CL)\\
&= G\boldsymbol{DI} && (A.4)\\
&= \boldsymbol{KKI}\,(\boldsymbol{DI}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{DI}) \equiv \boldsymbol{K}(\,\boldsymbol{S}(F(G\boldsymbol{I}))(\boldsymbol{KI})\boldsymbol{I}\,) && (CL)\\
&= \boldsymbol{K}(F(G\boldsymbol{I})\boldsymbol{I}\,(\boldsymbol{KII})\,) && (CL)\\
&= \boldsymbol{K}(\,F(G\boldsymbol{I})\boldsymbol{I}\,\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\,\boldsymbol{KSI}\,(G\boldsymbol{II})\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}\,(G\boldsymbol{II})\,\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(\,\boldsymbol{KKI}\,(\boldsymbol{II}))\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(\boldsymbol{K}\,(\boldsymbol{II})\,)\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(\boldsymbol{K}\,(\boldsymbol{KI}(\boldsymbol{KI}))\,)\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(\boldsymbol{KI})\boldsymbol{I}) \equiv \boldsymbol{K}(\boldsymbol{S}(\boldsymbol{K}(\boldsymbol{SKK}))\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(\,\boldsymbol{KKK}\,(\boldsymbol{SKK}))\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}\,(\boldsymbol{S}(\boldsymbol{KK})(\boldsymbol{SK})\boldsymbol{K})\,\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\,\boldsymbol{KSK}\,(\boldsymbol{S}(\boldsymbol{KK})(\boldsymbol{SK})\boldsymbol{K})\boldsymbol{I}) \equiv \boldsymbol{K}(\,\boldsymbol{KSK}\,(G(\boldsymbol{SK})\boldsymbol{K})\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\,\boldsymbol{S}(\boldsymbol{KS})(G(\boldsymbol{SK}))\boldsymbol{K}\,\boldsymbol{I}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(\boldsymbol{KS})(G(\boldsymbol{SK}))\boldsymbol{K}\,(\boldsymbol{KIK})\,) \equiv \boldsymbol{K}(F(G(\boldsymbol{SK}))\boldsymbol{K}\,(\boldsymbol{KIK})\,) && (CL)\\
&= \boldsymbol{K}(\,\boldsymbol{S}(F(G(\boldsymbol{SK})))(\boldsymbol{KI})\boldsymbol{K}\,) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(F(\,\boldsymbol{KGK}\,(\boldsymbol{SK})))(\boldsymbol{KI})\boldsymbol{K} && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(F\,(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{SK})\,)(\boldsymbol{KI})\boldsymbol{K}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}(\,\boldsymbol{KFK}\,(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{SK}))(\boldsymbol{KI})\boldsymbol{K}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{S}\,(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S})\boldsymbol{K})\,(\boldsymbol{KI})\boldsymbol{K}) && (CL)\\
&= \boldsymbol{K}(\,\boldsymbol{KSK}\,(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S})\boldsymbol{K})(\boldsymbol{KI})\boldsymbol{K}) && (CL)\\
&= \boldsymbol{K}(\,\boldsymbol{S}(\boldsymbol{KS})(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S}))\boldsymbol{K}\,(\boldsymbol{KI})\boldsymbol{K})\\
&\quad \equiv \boldsymbol{K}(\,F(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S}))\boldsymbol{K}\,(\boldsymbol{KI})\boldsymbol{K}) && (CL)\\
&= \boldsymbol{K}(F(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S}))\boldsymbol{K}\,(\boldsymbol{K}(\boldsymbol{KI})\boldsymbol{K})\,\boldsymbol{K}) && (CL)\\
&= \boldsymbol{K}(\,\boldsymbol{S}(F(\boldsymbol{S}(\boldsymbol{K}F)(\boldsymbol{S}(\boldsymbol{K}G)\boldsymbol{S})))(\boldsymbol{K}(\boldsymbol{KI}))\boldsymbol{K}\,\boldsymbol{K}) && (CL)\\
&= \boldsymbol{K}(\boldsymbol{SKK}) \equiv \boldsymbol{KI} && (A.2)
\end{aligned}
$$

Acknowledgements

We thank Henk Barendregt for stimulating discussions about this work, and Wojtek Swiatek, who in the framework of the MRI master class (UU-KUN, 1998–1999) composed the derivation in the Appendix, which is the crucial part of the proof of Theorem 8.4.

References

Asperti, A. and Ciabatoni, A. (1995) Effective applicative structures. In: Proceedings of the 6th biennial conference on Category Theory in Computer Science (CTCS'95). *Springer-Verlag Lecture Notes in Computer Science* **953** 81–95.

Asperti, A. and Ciabatoni, A. (1996) On completability of partial combinatory algebras. In: *Proceedings of the Italian Conference on Theoretical Computer Science, Ravello, 1995*, World Sci. Publishing, River Edge, NJ 162–175.

Asperti, A. and Ciabatoni, A. (1997) A sufficient condition for completability of partial combinatory algebras. *Journal of Symbolic Logic* **62** (4) 1209–1214.

Baader, F. and Nipkow, T. (1998) *Term rewriting and all that*, Research Notes in Theoretical Computer Science, Cambridge University Press.

Barendregt, H. P. (1971) On the interpretation of terms without a normal form. ERCU publicaties no. 111, Rijksuniversiteit Utrecht.

Barendregt, H. P. (1975) Normed uniformly reflexive structures. In: Böhm, C. (ed.) λ-Calculus and Computer Science Theory. *Springer-Verlag Lecture Notes in Computer Science* **37** 272–286.

Barendregt, H. P. (1984) *The Lambda Calculus. Its Syntax and Semantics* (second, revised edition), North-Holland, Amsterdam.

Bethke, I. (1987) On the existence of extensional partial combinatory algebras. *Journal of Symbolic Logic* **52** (3) 819–833.

Bethke, I. and Klop, J. W. (1996) Collapsing partial combinatory algebras. In: Dowek, G., Heering, J., Meinke, K. and Möller, B. (eds.) Higher-Order Algebra, Logic, and Term Rewriting (HOA '95). *Springer-Verlag Lecture Notes in Computer Science* **1074** 57–73.

Bethke, I., Klop, J. W. and de Vrijer, R. C. (1996) Completing partial combinatory algebras with unique head-normal forms. In: *Proc. 11th Annual IEEE Symposium on Logic and Computer Science, New Brunswick, New Jersey, 27–30 July 1996*, IEEE Computer Society Press 448–454.

Bethke, I., Klop, J. W. and de Vrijer, R. C. (1997a) Looking back. In: *Liber Amicorum, Paul Klint 25 jaar SMC/CWI*, CWI 43–55.

Bethke, I., Klop, J. W. and de Vrijer, R.C. (1997b) Origin tracking in orthogonal term rewriting systems. Technical Report IR-441, Vrije Universiteit Amsterdam, Department of Mathematics and Computer Science.

Curry, H. and Feys, R. (1958) *Combinatory Logic*, volume I, North-Holland, Amsterdam.

Hindley, J. R. (1980) Open problems; Swansea λ-calculus meeting, 21 September 1979. *Bulletin of the EATCS* **10**.

Klop, J. W. (1980) Combinatory Reduction Systems. *Mathematical Centre Tracts* **127**, Mathematisch Centrum, Amsterdam.

Klop, J. W. (1982) Extending partial combinatory algebras. *Bulletin of the European Association for Theoretical Computer Science* **16** 472–482.

Klop, J. W. (1992) Term rewriting systems. In: Abramsky, S. and Gabbay, D. (eds.) *Handbook of Logic in Computer Science, Volume 2*, Oxford University Press 1–116.

Lévy, J.-J. (1975) An algebraic interpretation of the $\lambda\beta k$-calculus and a labelled λ-calculus. In: Böhm, C. (ed.) λ-Calculus and Computer Science Theory, Proceedings of the Symposium held in Rome, March 25–27. *Springer-Verlag Lecture Notes in Computer Science* **37** 147–165.

Mitchell, J. (1996) *Foundations for Programming Languages*, MIT Press, Cambridge (MA).

Plotkin, G. D. (1985) Lectures on predomains and partial functions. Notes for a course given at the Center for the Study of Language and Information, Stanford.

Strong, H. R. (1968) Algebraically generalized recursive function theory. *IBM J. Research and Development* **12** 465–475.

Wagner, E. G. (1969) Uniformly reflexive structures. *Trans. American Math. Society* **144** 1–41.